PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-253882
(43)Date of publicati n of application : 03.10.1995

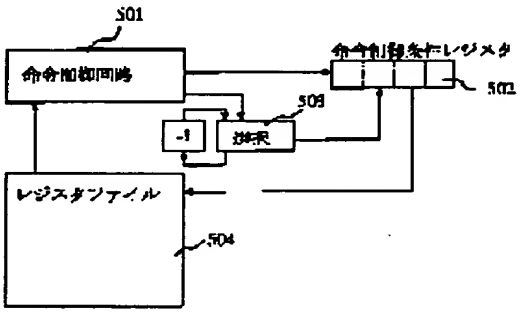
(51)Int.Cl. G06F 9/32
G06F 9/34

(21)Application number : 06-043529
(22)Date of filing : 15.03.1994

(71)Applicant : HITACHI LTD
(72)Inventor : NISHIYAMA HIROYASU

(54) COMMAND CONTROLLER

(57)Abstract:
PURPOSE: To realize the conditional execution function by adding commands which c ntrol the execution of the following command to the command controller according t the value of a register specified to the field in the command.
CONSTITUTION: The command which controls the execution of the following command is added to a command controller according to the value of the register specified to the field in the command. Then, when a command control circuit 501 of the command contr ller fetches a command, a register 503 for selecting conditions is checked. If the value is higher than 0, the value of a register 503 for selecting the condition is reduced by 1 and the field of a command control condition register 502 indicated by the register 503 is read out. Then, the register specified by the field of the register 502 is read out fr m the register file. The execution of the command is not performed when the value of the register is true and there is no specification for inverting the true or false value of th register or when the value of the register is false and there is a specification for inv rting the true or false value of the register.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number f appeal against examiner's decision of rejection]

[Dat of requesting appeal against examiner's decision of r jecti n]

[Dat f extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平7－253882

(43)公開日　平成７年(1995)10月３日

(51)Int.Cl.*		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ	9/32	３４０　Ｂ			
	9/34	３３０			

審査請求　未請求　請求項の数２　ＯＬ　（全 5 頁）

(21)出願番号　特願平6－43529

(22)出願日　平成６年(1994)３月15日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　西山　博泰
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(74)代理人　弁理士　小川　勝男

(54)【発明の名称】　命令制御装置

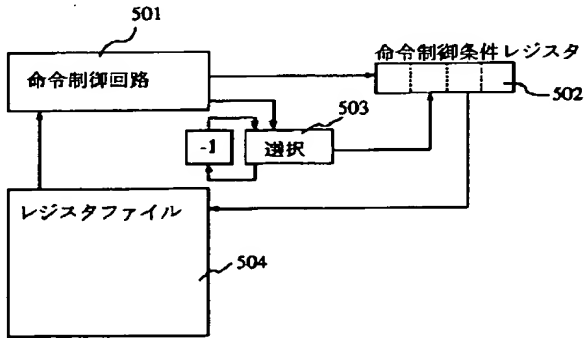
(57)【要約】

【目的】既存のアーキテクチャと命令互換性を保ったまま、ハードウェアに条件付き実行と多数のレジスタを参照する機能を追加して、効率良く実行させる。

【構成】命令内のフィールドに指定された複数個のレジスタの値によって、後続するの複数個の命令を実行するか否かを決定する命令と、命令内のフィールドに指定された複数個の値によって、後続する複数個の命令がレジスタファイル参照を行なう際の添字を変更することにより、命令フィールドで指定可能な数より多くのレジスタを参照できる命令とを、命令制御装置に付加する。

【効果】分岐命令の実行回数が減少し命令パイプラインの乱れが少なくなるので、実行性能が向上する。また、プログラムで参照できるレジスタが増加し、メモリ参照回数が減少するので、科学技術計算の高速化に有効である。

(図5)



1

【特許請求の範囲】

【請求項１】原始プログラムにおける条件分岐を含むコードを計算機で実行する場合に、命令内のフィールドに指定された複数個のレジスタの値によって、以降の複数個の命令を実行するか否かを決定する命令を有することを特徴とする命令制御装置。

【請求項２】多数のレジスタを有する演算装置に対して、命令内のフィールドにレジスタファイルを参照するためのオフセットを指定することで、以降の複数個の命令が参照するレジスタを変更することにより、命令フィールドで指定可能な数より多くのレジスタを参照することを可能とする命令を有することを特徴とする命令制御装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は科学技術計算を高速に実行するためのディジタル型電子計算機の命令制御装置に関する。

【０００２】

【従来の技術】科学技術計算を高速に実行するには、分岐命令の実行頻度を低下させることや、メモリの参照回数を減少させることが重要である。このためには、、条件付きの命令実行や、多数のレジスタを使用することを可能にすることが有効である。

【０００３】条件付き実行とは、各命令のオペランドフィールドで指定されたプレディケートと呼ばれるレジスタの値に従って命令を実行するか否かを決定する機能である。条件分岐命令の実行は命令パイプラインの乱れを発生し、命令実行速度を低下させるため、条件文を条件分岐命令の組合せではなく条件付き実行にコンパイルすることは、科学技術計算の高速な実行に効果がある。条件付き実行の機能は、従来はCydra 5(Rau, Yen, Yen, Towle: Cydra 5 Departmental Supercomputer,COMPUTER, Vol.22, No.1, IEEE)のような、命令語長が非常に長い多長命令(VLIW)方式の科学技術用計算機で用いられていた。命令語長が32ビットの既存アーキテクチャ上では、多長命令方式のようにプレディケートを指定するためのフィールドが存在しないため、これまでのところ既存アーキテクチャ上で条件付き命令実行をサポートする機構は採り入れられていない。

【０００４】既存アーキテクチャ上での条件付き実行と類似の機構としては、Hewlett Packard社のPA-RISC（PA-RISC 1.1 Architecture and Instruction Set Reference Manual, Hewlett Packard）で採り入れられている、命令の実行結果に従って次の1つの命令を実行を無効化する機構が存在する。

【０００５】多数のレジスタを使用可能とすることは、アクセス速度の遅いメモリとのやりとりを減少させることになり、多量のメモリを参照する科学技術計算の高速な実行に効果がある。多数のレジスタを使用することを

2

可能とする方式としては、上記Cydra 5で採用された回転レジスタファイルがある。回転レジスタファイルは、多数のレジスタを用いて、ループの各繰り返しに対してレジスタの集合を割り当て、これを切り替えながらループの処理を行なう機構である。回転レジスタファイルを持つアーキテクチャでは、各命令は、レジスタ番号と繰り返しオフセットを指定してレジスタファイルを参照する。回転レジスタファイルの場合も、命令語にレジスタ参照のための繰り返しのオフセットフィールドを付加する必要があるため、これまでのところ多長命令方式の計算機でのみ実現されている。

【０００６】命令語長が32ビットの既存アーキテクチャとの互換性を保った上で、多数のレジスタを有する回転レジスタと類似の効果を得る方式としては、スライドウィンドウ方式のレジスタウィンドウを用いたアーキテクチャ（位守、中村、朴、中澤：スライドウィンドウ方式による疑似ベクトルプロセッサ、情報処理学会論文誌、Vol.34, No.12, 1993年）と、レジスタコネクションによる方式（Kiyohara, Mahlke, Chen, Bringmann, Hank, Anik, Hwu: Register Connection: A New Approach to Adding Registers into Instruction Set Architectures, In Proceedings of The 20th International Symposium on Computer Architecture, 1993)が提案されている。スライドウィンドウ方式では、多数のレジスタのうち、命令が一度に参照できるレジスタを既存アーキテクチャと同じ個数とし、ウィンドウと呼ばれる命令から参照できるレジスタの集合を、ループの繰り返し毎に物理レジスタ上で移動することで回転レジスタと類似の機構を実現している。レジスタコネクションによる方式では、命令で参照可能なレジスタそれぞれに対して物理レジスタ番号との対応表を用意し、レジスタファイルを間接的に参照することで多数のレジスタを利用可能としている。

【０００７】

【発明が解決しようとする課題】条件付き実行は科学技術計算の高速な実行に有効な技術であるが、従来の方式では、命令にプレディケートを指定するフィールドを追加する必要があるため、既存アーキテクチャと命令互換性が保たれないという問題がある。 Hewlett Packard社のPA-RISCなどで採用されている、次の命令の実行を無効化する機構は、既存のアーキテクチャ上で、条件付き実行に類似した機能を実現するが、条件付きで実行したい命令1つにつき、これを実行するか否かを決定する命令を1つ必要とするため、命令数が増加し性能が低下してしまう。

【０００８】同様に、多数のレジスタを使用可能とすることは、科学技術計算の高速な実行に有効であり、その方式としては、回転レジスタファイルが知られている。回転レジスタをサポートするためには、命令に繰り返しのオフセットを示すフィールドを追加する必要があるた

め、既存アーキテクチャと命令互換性が保たれないという問題がある。

【０００９】スライドウィンドウ方式では、回転レジスタに類似した機能を提供するが、命令が一度に参照できるレジスタは限られているため、多重のレジスタの参照を必要とする問題では利用可能なレジスタが不足し、その場合には多数のレジスタがハードウェア上は存在するにも関わらず、低速なメモリへの参照を行う必要がある。また、レジスタコネクションによる方式では、各レジスタについて物理レジスタ番号との対応表を管理しなくてはならないため、ハードウェアの実現とソフトウェアによる利用が繁雑になる。

【００１０】本発明は以上のような課題に対して、既存のアーキテクチャのと互換性を保った上で、

1. レジスタの値に従って命令を実行するか否かを決定する命令制御装置と、

2. 多数のレジスタを参照できるようにする命令制御装置

とを提供することを目的とする。

【００１１】

【課題を解決するための手段】このため、本発明では、オペランドに指定されたレジスタの値に従って、後続する命令の実行を制御する命令（以下PRED命令とする）を有する命令制御装置により、既存のアーキテクチャと互換性を持つアーキテクチャ上で、条件付き実行と同様の機能を実現する。さらに、後続する命令のレジスタファイル参照のための添字を変更する命令（以下RR命令とする）を有する命令制御装置により、既存のアーキテクチャと命令互換性を持つアーキテクチャ上で、多数のレジスタを参照することを可能とする。

【００１２】各命令の動作の概要を述べる。命令制御ユニットでは以下のような処理を実行する：

1. PRED命令をフェッチすると、PRED命令のオペランドに指定された固定個のレジスタを記憶しておき、PRED命令に後続するi番目の命令をフェッチした際に、PRED命令のi番目のオペランドに指定されたレジスタの値に従って命令を実行するか否かを決定する。

【００１３】2. 多数のレジスタを持つハードウェアを持ち、レジスタファイル参照のための添字（以下カレントウィンドウポインタとする）からの相対位置でレジスタファイルの参照を行うアーキテクチャを仮定する。このアーキテクチャ上で、　ＲＲ命令をフェッチすると、ＲＲ命令のオペランドに指定された複数の値を記憶しておき、RR命令に後続するi番目の命令を実行する際に、RR命令のi番目のオペランドに指定された値をカレントウィンドウポインタの値に加え、この値をレジスタファイル参照のための添字としてレジスタファイルを参照し、命令を実行する。

【００１４】

【作用】手段1により、PRED命令のオペランドに指定したレジスタの値に従って、後続する命令の実行をするか否かを決定することが可能となる。これにより、既存のアーキテクチャにPRED命令を付加することで、条件付き実行と同様な機能が実現され、課題1を解決することができる。

【００１５】手段2により、RR命令に後続する命令を実行する際に、レジスタファイルを参照するための添字を、RR命令のオペランドで指定した値で変更することが可能となる。この機能により、既存のアーキテクチャと命令互換性を守った上で、多数のレジスタを参照する機能が実現され、課題2を解決することができる。

【００１６】以上の手段により、既存アーキテクチャと命令互換性を保った上で、条件付き実行と多数のレジスタを参照するための機構とを実現することが可能となり、これにより科学技術計算を高速に実行することができる。

【００１７】

【実施例】以下、本発明の実施例を図表を参照しつつ説明する。本実施例では、通常の RISC型プロセッサと同じく、命令幅が32ビット、命令フィールドで指定可能なレジスタが32個のアーキテクチャを想定する。

【００１８】まず、PRED命令の1つの実施例を説明する。ここでは、1つのPRED命令で後続する４つの命令を実行するか否かを制御できるものとし、さらに、レジスタの値の真偽を反転するか否かを指定できるものとする。このため、PRED命令は4つのオペランドをとり、それぞれのオペランドは命令を実行するか否かを決定するレジスタと、レジスタの値の真偽を反転するか否かを指定するものとする。なお、5つ以上のオペランドに対しても同様に実現できる。

【００１９】図１はPRED命令の形式の例である。この図で101はPRED命令の命令コードを示す8ビットのフィールドである。102以降の4つのフィールドは、PRED命令に後続する4つの命令を制御するレジスタとその値の真偽を反転するか否かを指定する6ビットのフィールドで、6ビットのフィールドのうち、5ビットで32個のレジスタを指定し、残る1ビットで値の真偽を反転するか否かを指定する。 命令制御装置は、PRED命令に後続する4つの命令うち、i番目の命令をフェッチした際に、PRED命令のi番目のフィールドで指定されたレジスタを参照し、レジスタの値が真かつ真偽の反転が指定されていないか、レジスタの値が偽かつ真偽の反転が指定されている場合に、命令の実行を行わないようにする。

【００２０】図２は分岐を含む原始プログラムの例である。ここで、QとRは定数とする。このプログラムを従来のアーキテクチャ向けにコンパイルした結果を図３に示す。次に、図２のプログラムをPRED命令を用いたアーキテクチャ向けにコンパイルした結果を図４に示す。図４のコンパイル結果では、命令401によって命令402、命令403、命令404、命令405を実行するか否かを決定するレ

ジスタを指定している。ここでは、PRED命令401の1番目と4番目のオペランドにレジスタTを指定し、2番目のオペランドにレジスタp、3番目のオペランドにレジスタpの真偽を反転したものを指定している。ここに示すように、PRED命令のオペランドにおいて、レジスタの反転の指定は!で表すものとする。レジスタTの値は常に真であるものとすると、命令402と命令405は常に実行され、命令403 はレジスタpの値が真の場合、命令404はレジスタpの値が偽の場合に実行される。

【００２１】図３のコンパイル結果では、命令301と命令302の分岐命令の数だけ命令数が多くなっており、かつ、分岐の実行を行なうと命令実行パイプラインに乱れを生じるため、図４のプログラムは図３のプログラムと比較して、高速に実行することができる。

【００２２】図５はPRED命令を実現するためのハードウェアの例である。命令制御回路 501は、PRED命令をフェッチすると、オペランドフィールドに指定されたレジスタの番号とレジスタの真偽を反転するかどうかの指定を、逆順に命令制御条件レジスタ502に記憶し、条件選択用のレジスタ503の値を、PRED命令が制御する命令の数に設定にする。命令制御回路501が命令をフェッチする場合には、条件選択用のレジスタ503を調べ、その値が0以上であれば、条件選択用のレジスタ503の値を1減じ、条件選択用のレジスタが指している命令制御条件レジスタ502のフィールドを読みだす。次に、命令制御条件レジスタのフィールドで指定されたレジスタをレジスタファイル504から読みだし、レジスタの値が真かつレジスタの真偽値を反転する指定がないか、レジスタの値が偽かつレジスタの真偽値を反転する指定がある場合に、命令の実行を行なわない。

【００２３】続いて、RR命令の1つの実施例を説明する。ここでは、ハードウェアは128個のレジスタからなるレジスタファイルを持つものとし、通常の命令によるレジスタの参照はスライドウィンドウ方式のレジスタウィンドウにより、カレントウィンドウポインタの指す位置から32個のレジスタを参照できるものとする。 RR 命令は4つのオペランドをとり、後続する命令がソースレジスタを参照する際のレジスタファイル参照のオフセット値を指定するものとする。

【００２４】図６はRR命令の形式の例である。この図で601はRR命令の命令コードを示す 4ビットのフィールドである。602以降の4つのフィールドは-127から0までの値をとる7ビットのフィールドからなり、レジスタファイルを参照する際のオフセットを指定する。命令制御装置が、RR命令に後続する4つの命令のうちi番目の命令を実行する場合、カレントウィンドウポインタにRR命令のi番目のフィールドで指定された値を加えて、ソースレジスタに対するレジスタファイルの参照を行なう。ここでは、ターゲットレジスタについては、レジスタファイル参照のための添字の変更は行なわないものとする。

【００２５】図７はDO文を含む原始プログラムの例である、ここで、Qは定数値とする。これを従来のアーキテクチャ向けにコンパイルした例を図８に示す。図８のコードでは、ロード命令801によって配列の要素x[i-50]をメモリからロードしている。この値は加算命令802によって50回前の繰り返しで計算したものであり、多数のレジスタが利用可能であればレジスタに保存しておけばよく、メモリから再度読み出す必要はない。次に、図７のコードを本方式を用いたアーキテクチャ向けにコンパイルした結果を図９に示す。図９のプログラムでは、レジスタウィンドウのスライドを行なう命令906によって各繰り返し毎にレジスタウィンドウを2つづつスライドさせており、レジスタt0およびt1は繰り返し毎に異なる物理レジスタを指すことになる。RR命令901では、最初のオペランドに-100を指定することにより、RR命令901に後続する最初の命令902でのソースレジスタt1の参照が、50回前の繰り返しで命令903により計算したt1の値となるようにしている。命令901の2番目から 4番目までのオペランドには0を指定しているため、命令903から命令905までの命令のソースレジスタ参照のための添字は変更されないで実行される。このように、図８の例で必要となっているメモリから値の再ロードを行なう命令が、RR命令を用いることにより図９の例では不要となっている。既存アーキテクチャにスライドウィンドウを適用しただけでは、一度に参照可能なレジスタが限られるため、図８の例のようにメモリから再ロードしなければならない。

【００２６】図１０はRR命令を実現するためのハードウェアの例である。命令制御回路 1001は、RR命令をフェッチすると、オペランドに指定された4つの値をオフセット保存レジスタ1002に逆順に記録し、オフセット選択用のレジスタ1003の値を、 RR命令が制御する命令の数に設定にする。命令制御回路1001が命令を実行する場合には、オフセット選択用レジスタ1003の値を調べ、その値が0以上であれば、オフセット選択用レジスタ1003の値を1減じ、オフセット選択用レジスタ1003が指しているオフセット保存レジスタの値を、カレントウィンドウポインタ1004の値に加え、この値にレジスタ番号を加えた値をレジスタウィンドウの添字として、レジスタファイル1005からソースレジスタを読み出し、実行ユニット1006によって実行を行なう。

【００２７】なお、本実施例ではRR命令ではソースレジスタのレジスタファイル参照の添字のみを変更するものとしたが、この他にも、デスティネーションレジスタのみを変更したり、ソースレジスタ、デスティネーションレジスタの両方を変更するなど幾つかの実施方式を考えることができる。

【００２８】

【発明の効果】本発明によれば、既存のアーキテクチャと互換性を保ったままで、条件付き実行と多数のレジス

タを参照するための機能を付け加えることができる。条件付き実行は、プログラム中の条件分岐命令を不要にし、条件分岐のための命令数の削減や、命令パイプラインの乱れを除去する効果があり、科学技術計算の高速化に効果がある。同様に、多数のレジスタを参照可能にすることは、低速なメモリとのデータ転送回数を減少させ、科学技術計算の高速化に効果がある。

【図面の簡単な説明】

【図１】PRED命令の命令形式の説明図である。

【図２】条件文を含んだ原始プログラムの例である。

【図３】図２の原始プログラムを従来のアーキテクチャ用にコンパイルした場合のコードの例である。

＊【図４】図２の原始プログラムをPRED命令を用いてコンパイルした場合のコードの例である。

【図５】PRED命令を実現するためのハードウェアの構成図である。

【図６】RR命令の命令形式の説明図である。

【図７】DO文を含んだ原始プログラムの例である。

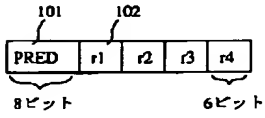
【図８】図７の原始プログラムを従来のアーキテクチャ用にコンパイルした場合のコードの例である。

【図９】図７の原始プログラムをRR命令を用いてコンパイルした場合のコードの例である。

【図１０】RR命令を実現するためのハードウェアの構成図である。

【図１】

(図1)



【図２】

(図2)

```
IF p THEN
  v = a[i]*Q
ELSE
  v = a[i]+R
b[i] = v
```

【図３】

(図3)

```
LOAD a[i],t
BF p, label1
MPY t,Q,v
B label2
label1:
  ADD t,R,v
label2:
  STORE v,b[i]
```

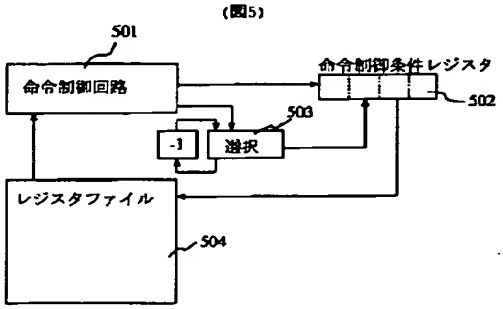
【図４】

(図4)

```
PRED T, p, !p, T
LOAD a[i],t
MPY t,Q,v
ADD t,R,v
STORE v,b[i]
```

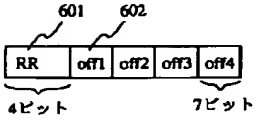
【図５】

(図5)



【図６】

(図6)



【図７】

(図7)

```
DO 100, I=0, 1000
  x[i] = x[i] +x[i-50]*Q
END DO
```

【図８】

(図8)

```
LOOP:
  LOAD x[i], t0
  LOAD x[i-50], t1
  MPY t1,Q,t1
  ADD t0,t1,t1
  STORE t1,x[i]
  ADD i,1,i
  if(i <= 1000) goto LOOP
```

【図９】

(図9)

```
LOOP:
  LOAD x[i], t0
  RR -100,0,0,0
  MPY t1,Q,t1
  ADD t0,t1,t1
  STORE t1,x[i]
  ADD i,1,i
  SLIDE 2
  if( i <= 1000) goto LOOP
```

【図１０】

(図10)